Distributed High Performance WEB CRAWLERS

Ms. PAYAL V. PATEL
Computer Engineering Department
Birla Vishvakarma Mahavidyalaya Engineering College
Vallabh Vidyanagar,
Anand, India

Prof. MOSIN I. HASAN
Computer Engineering Department
Birla Vishvakarma Mahavidyalaya Engineering College
Vallabh Vidyanagar,
Anand, India

Prof. BHAVESH A. TANAWALA
Computer Engineering Department
Birla Vishvakarma Mahavidyalaya Engineering College
Vallabh Vidyanagar,
Anand, India

Abstract: **Single crawlers are no longer sufficient to run on the web efficiently as explosive growth of the web pages. So, distributed crawlers come into the picture. Here we present a distributed high performance crawler which is used for crawling internet web pages. In that List Server, Crawler Manager and Crawlers are there. At the crawler side Yioop! is going to be used. Yioop! is php search engine which produces the index of web page. List Server is useful for maximizing the efficiency. The Crawler Manager can easily communicate with the different List Server. Likewise, the system will work like focused crawler as well as distributed crawler. The architecture of the system and the function of every module are described in detail which can be extended to other fields easily.**

Keywords: List Server, Distributed crawler, Crawler Manager, Yioop!

I. INTRODUCTION

Now a day in this competitive world timely information retrieval is a solution for survival. For the entire web, web crawlers provide searching and indexing support. Crawlers are programs which traverse through the web searching for the relevant information using algorithms that narrow down the search by finding out the most closer and relevant information [1].

Web crawling is a computationally expensive process that demands large amount of resources, including CPU, disk storage, memory, and network. Consequently, if the target is to crawl a significant portion of the web the single processor crawler systems will not succeed to achieve this. So instant solution is there to employ parallelization and perform crawling on multiple processors. However it is not possible to obtain the maximum benefit and achieve scalability in terms of network if all processors are located at a single data center. On the other hand geographically distributed web crawling contains multiple parallel crawlers and are situated at geographically distant data centers. Here, the potential benefit is that information about the spatial locality of the web servers can be used to achieve scalability and to improve the download time on the network resource as well.

1.1 WEB CRAWLER STRATEGIES

In web crawler there are two basic approaches: Blind Traversing Approach and Best First Heuristic Approach. For the implementation of web crawler many different algorithms are there. Here we describe brief overview of two algorithms from them.

1. Breadth First Search Algorithm

This algorithm aims in the uniform search across the neighbor nodes. It starts at the root node and searches the all the neighbor nodes at the same level. If the objective is reached then it is reported as success and the search is terminated otherwise it proceeds down to the next level sweeping the search across the neighbor nodes at that level and so on until the objective is reached. When all the nodes are searched and the objective is not met then it is reported as failure [2]. It will not perform so well when the branches are so many in a tree. For example: Chess game. When the entire path leads to the same objective with the same length of the path at that time this will not perform so well [3].

2. Page Rank Algorithm

Page rank algorithm determines the importance of the web pages by counting citations or back links to a given page. The page rank of a given page is calculated as

$$PR(A) = (1-d) + d\ (PR(T1)/C(T1) + \ldots + PR(Tn)/C(Tn))$$

Where, PR(A) is the Page Rank of page A.

PR(Ti) is the Page Rank of pages Ti which link to page A.

C(Ti) is the number of outbound links on page Ti and

d is a damping factor which can be set between 0 and 1.

Shaojie Qiao as in [4] proposed a new page rank algorithm. A new page rank algorithm is based on the correspondence measure from the vector space model. They proposed a new correspondence measure for computing the similarity of pages and apply it to division of web database into several web social networks.

II. CRAWLING TECHNIQUES

A general purpose web crawler gathers as many pages as it can from a particular set of URLs'. In that some crawler are designed to collect documents only on a specific topic such as for example for education related documents, for entertainment related documents etc. So these types of crawlers are called as Focused Crawler.

1. Focused Crawling

A general purpose web crawler gathers as many pages as it can from a particular set of URLs'. Where as a focused crawler is designed to collect the documents on a specific topic as a result that will reduce the amount of network load and download. The objective of the focused crawler is to selectively search for pages that are relevant to a predefined set of topics. This leads to significant savings in hardware and network resources. The topics are specified not using keywords but using commendable documents. Rather than collecting and indexing all accessible web credentials this crawler diagnosis its crawl boundary to search different URLs. This links are most significant for the crawl. That will avoid irrelevant regions of the web [5]. The most important evaluation of focused crawling is to measure the return ratio which is rate at which relevant pages are acquired and irrelevant pages are efficiently filtered off from the crawl. The focused crawler would spend a lot of time if this return ratio is low. For simply eliminating irrelevant pages and it may be better to use an ordinary crawler in its place.

2. Distributed Crawling

As the size of the web is increasing it has become very important to parallelize the crawling process to finish downloading the pages in a reasonable amount of time. A single crawling process will be insufficient for large scale engines that need to fetch large amounts of data rapidly even if multi threading is used. When a single centralized crawler is used all the fetched data passes through a single physical link. Distributing the crawling activity via multiple processes can help to build scalable system [6]. By splitting the load we can decrease hardware requirements and at the same time increase the overall download speed and reliability. Each task is performed in a fully distributed fashion means no central coordinator exists.

III. DISTRIBUTED CRAWLER

Centralized crawlers are no longer sufficient to crawl huge information. As the size of URL set grows, it is very important to design distributed crawler to parallelize the crawling process. The distributed crawler means several crawlers crawl information simultaneously on different computers. The distributed crawler system usually has a host computer as a coordinator called crawl manager which manages and distributing URLs to different crawlers [8]. There are mainly two categories of distribution strategies.

Static distribution strategy: In Static distribution strategy URLs set is divided into N (N is the number of crawlers in the system) subsets before crawling. In that each crawler crawls only one subsets.

Dynamic distribution strategy: In Dynamic distribution strategy URLs set are divided into M subsets where M>N. In that Crawler Manager first distributes one subset to each crawler. After that when one crawler has crawled all the seeds in the subset then the Crawler Manager will distribute another subset to it. The resource can be utilized appropriately in this strategy [2].

In the crawling process, many link URLs are extracted from web pages and some of them may be obtained by more than one crawler. The general URL assignment algorithm is the algorithm based on hash [9]. The Crawler Manager distributes URLs to crawlers dynamically and adjusts the size of the task packets flexibly according to different crawling period of each website.

3.1 THE ARCHITECTURE OF DISTRIBUTED CRAWLER

The architecture is shown in Figure 1. There are mainly three parts: the Crawler Manager, the set of crawlers and the crawling results server [10].

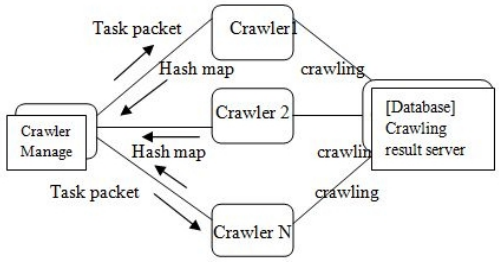

Figure 1 System Architecture of Distributed Crawler.

1) Crawler Manager

It is the central part of the whole distributed system. The Crawler Manager not only manages the seeds set needed to be crawled but also monitors all the crawlers' running states. During the crawling process the Crawler Manager loads all the URLs from seeds file, all the templates from templates file and all the hash map records into memory.

2) Crawlers

The crawlers in a distributed system execute a crawling analyzing save process. Distributed crawlers must negotiate with Crawler Manager including sending state information and hash maps to Crawler Manager and getting task packets from Crawler Manager. Task packet is the most important part. Task packet contains seeds, templates and hash map records.

3) Crawling results server.

Crawling results server is a storage system. The valuable information obtained by the crawlers will be sent to it and indexed in database for search.

IV. THE PROPOSED ARCHITECTURE OF DISTRIBUTED CRAWLER

In the architecture of the distributed crawler, Crawler Manager is there to distribute seeds and mean while monitor the state of each crawler. The architecture is shown in Figure 2. There are mainly three parts: the Crawler Manager, the set of crawlers and set of List Servers.

1) Crawler Manager

Crawler Manager takes the seeds from the particular List Server and then distributes seeds to it. Suppose one crawler is work for education related data. The Crawler Manager will take the seeds from that List Server which contains the education related URLs. After that Crawler Manager will distributes that URLs information to that crawler.

2) Crawler

Crawlers are the task executants of the system. The crawlers in a distributed system execute a crawling analyzing save process. Distributed crawlers must negotiate with Crawler Manager and getting URLs from Crawler Manager. Here Yioop! is used for crawling process and as a crawler. The Yioop! Search engine can produce indexes of web pages or a set of web pages whose total number of pages are in the tens of millions. It allows users to control the sites that should be indexed and hence gives control over the search results that will be returned. In Yioop! Directory there is bin folder which has the fetcher.php and queue_server.php which are used to start a crawl. The queue_server.php is responsible for maintaining the list of URLs to be crawled and also information about which sites have been crawled where as the fetcher is responsible for actually downloading the web pages provided by the queue_server[11].

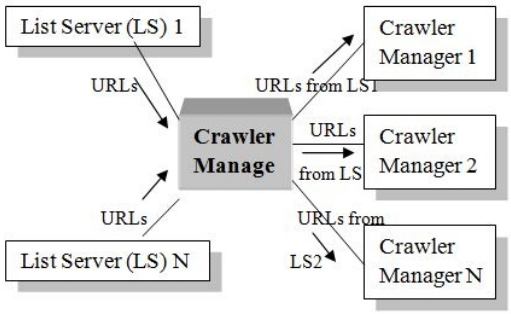

Figure 2 Proposed system architecture of distributed crawler.

3) List Servers

For large amount of websites Crawler Manager will spend more time than the estimated longest waiting time for crawler. So the List Server is useful for maximizing the efficiency. Different List Server contains particular topic related contents like only education related documents or entertainment related links etc. Our crawler will allow to crawl the particular topic related contents like only education related documents or entertainment related links etc. So by this way our crawler will work like distributed crawler.

To maximize the efficiency gain of the crawling system a third part website list service is an efficient way. Each website will be able to register itself on List Server as shown in Figure 3. So Crawler Manager can easily maintain its websites list by communicating with List Server.

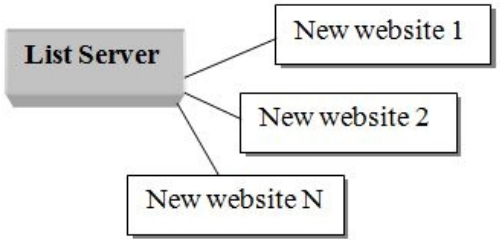

Figure 3 Website Registration

The Crawler Manager can easily communicate with the List Server. Each List Server contains different topics related data like education related data, entertainment related data etc. For the communication between List Server and Crawler Manager IP address and port no is required.

V. IMPLEMENTATION

Now, we describe implementation part of the system. Here it shows the information about implementation of all three modules.

5.1 LIST SERVER IMPLEMENTATION

List Server contain different URLs list. One List Server contains only education related URLs while other List Server contains only entertainment related URLs. For implementing List Server we are using XML files. One XML file contains only education related URLs list while another XML file contains entertainment related URLs list. At the time of new website registration we have to select particular category in which our new website will locate. After clicking on submit button "successfully registered" message will appear. This message shows that new website has been registered with particular List Server. New website will be added to the particular XML file by selecting the category element from the dropdown list box. Here, to register the new website we have to check the condition from dropdown list box which category has been selected based on that we have to add that new website to the XML file. Here we have to specify the location of the XML file. Different List Server can easily communicate with the Crawler Manager.

5.2 CRAWLER MANAGER IMPLEMENTATION

Crawler Manager reads the URLs from the particular List Server and submits them to the particular Yioop!, which contains information about seed sites. So our Crawler Manager distributes URLs to the particular Yioop!. For the implementation of Crawler Manager we have to use web service. Web services are web based applications implemented with an evolving set of open standards that enable application programs on various computers to communicate with other application programs on similar or disparate computers transparently over the Internet [12]. For reading URLs from the particular List Server we have to specify the location of that List Server. Here for implementing List Server we are using XML file.

5.3 YIOOP ATTACHMENT MODULE

Yioop! contains the information about seed sites. The seed site allows you to specify a list of URLs that the crawl should start from there. The crawl will begin using these URLs. Crawler Manager reads the URLs from the particular List Server and submits them to the particular Yioop!, which contains information about seed sites. So our Crawler Manager distributes URLs to the particular Yioop! For this, in Yioop! Crawler we have to modify the file that contains the information about the seed URLs. Yioop! has two files that contains seed URLs information. O1ne to read in crawl options either from the default_crawl.ini file or from the crawl options used in a previous crawl. For the first time crawler will read seed URLs from the default_crawl.ini file. After that it will read URLs from the crawl.ini file which will be located in the work directory. So here we have to put URLs list either in the default_crawl.ini file or in crawl.ini file. So our Crawler Manager distributes URLs to the particular Yioop! crawler and that crawler will work like focused crawler. Likewise different focused crawlers make distributed high performance web crawler together.

VI. CONCLUSION

This distributed crawler has been used to collect the Internet forum information and performed well. In fact, this crawler can be easily extended to collect blogs, news or some special information such as house rent information, traveling information and so on. So by this way crawler will work like focused crawler as well as distributed crawler.

REFERENCES

[1] Pavalam S M, Jawahar M, Felix K Akorli, S V Kashmir Raja on "Web Crawler in Mobile Systems" [International Conference on Machine Learning, ICMLC 2011].

[2] Steven S. Skiena on "The Algorithm design Manual" 2nd ed, Springer: London Limited, Pg 162, 2008.

[3] Ben Coppin on "Artificial Intelligence illuminated" Jones and Barlett Publishers, Pg 77, 2004

[4] Shaojie Qiao, Tianrui Li, Hong Li and Yan Zhu, Jing Peng, Jiangtao Qiu "SimRank: A Page Rank Approach based on similarity measure"[IEEE, 2010].

[5] Brin, Sergey and Page Lawrence on "The anatomy of a large-scale hyper textual Web search engine". Computer Networks and ISDN Systems, April 1998.

[6] http://www.webreference.com

[7] Gao Qing, Xiao Bo, Lin Zhiqing, Chen Xiyao, Zhou Bing on "A High-precision Forum Crawler Based on Vertical Crawling".[Proceedings of the 2009 International Conference on Network Infrastructure and Digital Content].

[8] Vladislav Shkapenyuk, Torsten Suel on "Design and Implementation of a High-Performance Distributed Web Crawler".[Proceedings of the 18th International Conference on Data Engineering].

[9] Yuan Wan, Hengqing Tong on "URL Assignment Algorithm of Crawler in Distributed System Based on Hash",[IEEE International Conference on Networking, Sensing and Control, 2008].

[10] Bing Zhou, Bo Xiao, Zhiqing Lin, Chuang Zhang on "A Distributed Vertical Crawler Using Crawling-Period Based Strategy",[2nd International Conference on Future Computer and Communication,2010].

[11] Yioop!Documentation.(n.d.) Retrieved Jan 10, 2011 from SeekQuarry webpage:http://seekquarry.com

[12] https://www.ischool.utexas.edu/~i385f/archive/liaw_h/http-webpaper.htm

